

# Open Arcade Architecture Device (OAAD) OAArcade Library Specification

Rev. 1.000

November 30, 1998

By: Burt Bicksler and Christopher D. Watkins

Industrial Mindworks, Inc.

<http://www.industrialmindworks.com/>

Copyright © 1998, Intel Corporation. All rights reserved. Permission must be granted by the author for reprint or use.

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

A LICENSE IS HEREBY GRANTED TO REPRODUCE AND DISTRIBUTE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED OR INTENDED HEREBY.

AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE SUCH RIGHTS.

All product names are trademarks, registered trademarks, or service marks of their respective owners.

*Please send comments via electronic mail to [Mark\\_Atkins@ccm.intel.com](mailto:Mark_Atkins@ccm.intel.com)*

## Revision History:

1.000	11/30/98	Initial Version
-------	----------	-----------------

## Table of Contents

<b>TABLE OF CONTENTS .....</b>	<b>2</b>
<b>1 INTRODUCTION .....</b>	<b>3</b>
<b>2 FUNCTIONS .....</b>	<b>3</b>
2.1 <i>oaaInit</i> .....	3
2.2 <i>oaaDeInit</i> .....	3
2.3 <i>oaaNumDevices</i> .....	4
2.4 <i>oaaGetDeviceName</i> .....	4
2.5 <i>oaaOpenDevice</i> .....	4
2.6 <i>oaaCloseDevice</i> .....	4
2.7 <i>oaaNumOfDigitalInputs</i> .....	5
2.8 <i>oaaNumOfDigitalOutputs</i> .....	5
2.9 <i>oaaNumOfCoinDoors</i> .....	5
2.10 <i>oaaGetCoinCount</i> .....	6
2.11 <i>oaaGetCoinStartCount</i> .....	6
2.12 <i>oaaGetCoinServiceCount</i> .....	6
2.13 <i>oaaGetCoinTestCount</i> .....	7
2.14 <i>oaaGetCoinTiltCount</i> .....	7
2.15 <i>oaaGetDigitalInput</i> .....	7
2.16 <i>oaaSetDigitalOutput</i> .....	8
<b>3 EXAMPLE .....</b>	<b>8</b>
<b>4 BUILDING AN APPLICATION .....</b>	<b>9</b>
<b>5 LIMITATIONS .....</b>	<b>9</b>

## 1 Introduction

This document describes the OAARcade Library that wraps a subset of the functionality of the OAAO COM based Library. The OAARcade Library simplifies the task of adding coin door and digital input/output functionality to an arcade application. Any application that can call a Windows DLL can use this library.

The OAARcade Library is provided in binary form, with Debug and Release builds for VC++ 5.0. There is a sample application provided in source and binary form that illustrates how to use the functionality provided in the library wrapper. The sample application name is OAARcadeExamp.

For the full functionality of the OAAO Library refer to the GUIExample application and the OAAO Library documentation.

## 2 Functions

### 2.1 oaaInit

Load the OAAO Library, initialize any variables that are required.  
Performs an internal enumeration of the devices found on the system.

This must be the first OAARcade function that is called.

Parms:

None.

Returns:

OA\_OK or error code.

HRESULT oaaInit() ;

### 2.2 oaaDeInit

Release the OAAO Library, release any resources that were allocated.  
This must be done after all other access to the library is completed  
and before the application terminates.

This must be the last OAARcade function that is called, and no more  
calls are allowed to this library after this function has been called.

Parms:

None.

Returns:

OA\_OK or error code.

HRESULT oaaDeInit() ;

## 2.3 oaaNumDevices

Returns the number of devices that were located during initialization.

Parms:

pnCount                      Pointer to an integer that receives the number of devices.

Returns:

OA\_OK or error code.

HRESULT oaaNumDevices( int \*pnCount ) ;

## 2.4 oaaGetDeviceName

Returns the number of devices that were located during initialization.

Parms:

ppszDeviceName              Pointer to pointer to a STR that receives the name of the requested device.

nIndex                        Integer index, 0 based, to the requested device.

Returns:

OA\_OK or error code.

HRESULT oaaGetDeviceName( int nIndex, LPSTR \*ppszDeviceName ) ;

## 2.5 oaaOpenDevice

Opens the requested device and returns a handle to it.

Parms:

pszDeviceName      Pointer to the Device Name returned by a previous call to oaaGetDeviceName().

pDeviceHandle      Pointer to a HANDLE that receives the handle that identifies the requested device. If \*pDeviceHandle is NULL then there was a failure to open the device.

Returns:

OA\_OK or error code. If OA\_OK then \*pDeviceHandle will contain the handle that should be used in subsequent calls to the library for this device.

HRESULT oaaOpenDevice( LPSTR pszDeviceName, HANDLE \*pDeviceHandle ) ;

## 2.6 oaaCloseDevice

Closes the requested device. This function should be called when the application is

finished with the device. If this function is not called before the application exits there will be resource leaks, as the OAAO Device COM Object will not have been released.

**Parms:**

hDevice	A Handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.
---------	---

**Returns:**

OA\_OK or error code.

HRESULT oaaCloseDevice( HANDLE hDevice ) ;

## 2.7 oaaNumOfDigitalInputs

Returns the number of General Purpose Digital Inputs supported by the requested device.

**Parms:**

hDevice	A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.
pDigInputs	Pointer to an integer that receives the number of digital inputs on the requested device.

**Returns:**

OA\_OK or error code.

HRESULT oaaNumOfDigitalInputs( HANDLE hDevice, int \*pDigInputs ) ;

## 2.8 oaaNumOfDigitalOutputs

Returns the number of General Purpose Digital Outputs supported by the requested device.

**Parms:**

hDevice	A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.
pDigInputs	Pointer to an integer that receives the number of digital outputs on the requested device.

**Returns:**

OA\_OK or error code.

HRESULT oaaNumOfDigitalOutputs( HANDLE hDevice, int \*pDigOutputs ) ;

## 2.9 oaaNumOfCoinDoors

Returns the number of Coin Doors supported by the requested device.

**Parms:**

hDevice	A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.
---------	---

**pDigInputs**      Pointer to an integer that receives the number of Coin Doors on the requested device.

Returns:

OA\_OK or error code.

HRESULT oaaNumOfCoinDoors( HANDLE hDevice, int \*pCoinDoors ) ;

## 2.10 oaaGetCoinCount

Returns the number of coin drops on the designated coin door on the requested device.

Parms:

**hDevice**      A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.

**nCoinDoorNum**    The 0 based index to the requested coin door.

**\*nCount**      Pointer to an integer that receives the requested count. The count is returned as a modulo-255 count. The application needs to compare the current count with the new count. If the new count is less than the previous count then the application should assume that the counter has wrapped, and make the appropriate adjustment to the total coin count that the application is maintaining.

Returns:

OA\_OK or error code.

HRESULT oaaGetCoinCount( HANDLE hDevice, int nCoinDoorNum, int \*nCount ) ;

## 2.11 oaaGetCoinStartCount

Returns the current Coin Start count for the designated Coin Door on the requested device.

Parms:

**hDevice**      A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.

**nCoinDoorNum**    The 0 based index to the requested Coin Door.

**\*pCoinStart**      Pointer to an integer that receives the current Start count.

Returns:

OA\_OK or error code.

HRESULT oaaGetCoinStartCount( HANDLE hDevice, int nCoinDoorNum, int \*pCoinStart ) ;

## 2.12 oaaGetCoinServiceCount

Returns the current Coin Service count for the designated Coin Door on the requested

device.

Parms:

hDevice            A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.

nCoinDoorNum    The 0 based index to the requested Coin Door.

\*pCoinService    Pointer to an integer that receives the current Service count.

Returns:

OA\_OK or error code.

HRESULT oaaGetCoinServiceCount( HANDLE hDevice, int nCoinDoorNum, int \*pCoinService );

### 2.13 oaaGetCoinTestCount

Returns the current Coin Test count on the requested device.

Parms:

hDevice            A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.

\*pCoinTest        Pointer to an integer that receives the current Test count.

Returns:

OA\_OK or error code.

HRESULT oaaGetCoinTestCount( HANDLE hDevice, int \*pCoinTest );

### 2.14 oaaGetCoinTiltCount

Returns the current Coin Tilt count on the requested device.

Parms:

hDevice            A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.

\*pCoinTilt        Pointer to an integer that receives the current Tilt count.

Returns:

OA\_OK or error code.

HRESULT oaaGetCoinTiltCount( HANDLE hDevice, int \*pCoinTilt );

### 2.15 oaaGetDigitalInput

Returns the designated Digital Input count value on the requested device.

Parms:

hDevice            A handle to the requested device. This handle was returned by a call to

the oaaOpenDevice() function.

nInputNum	The 0 based index to the requested Digital Input.
*nCount	Pointer to an integer that receives the current Digital Input count for the requested input.

Returns:

OA\_OK or error code.

```
HRESULT oaaGetDigitalInput( HANDLE hDevice, int nInputNum, int *nCount ) ;
```

## 2.16 oaaSetDigitalOutput

Sets/Clears the designated Digital Output on the requested device.

Parms:

hDevice	A handle to the requested device. This handle was returned by a call to the oaaOpenDevice() function.
nOutputNum	The 0 based index to the requested Digital Output.
bSet	A Boolean that sets the designated output to 1 (TRUE) or 0 (FALSE).

Returns:

OA\_OK or error code.

```
HRESULT oaaSetDigitalOutput( HANDLE hDevice, int nOutputNum, BOOL bSet ) ;
```

## 3 Example

Here is a very simple example of how to use the OAArcade Library.

```
#include "OAArcadeIF.h"
```

```
HANDLE hDevice = NULL ;
```

```
void main()
```

```
{
    HRESULT hr = oaaInit() ;
    If( FAILED( hr ) )
        ... Handle error here.
    int nNumDevices = 0 ;

    hr = oaaNumDevices( &nNumDevices ) ;
    if( FAILED( hr ) )
        ... Handle error here.

    LPSTR pszDeviceName = NULL ;
    for( int i = 0 ; i < nNumDevices; i++ )
    {
```



```

        hr = oaaGetDeviceName( i, &pszDeviceName ) ;
        ... error handling here, add name to local list to present to user/operator.
    }

    /*
    *      Here we need to open the device that the user has selected.
    */
    char szSelectedDeviceName[512] ;
    /* set szSelectedDeviceName to the name the user selected. */
    hr = oaaOpenDevice( szSelectedDeviceName, &hDevice ) ;
    ... error handling here.

    int nNumCoinDoors = 0 ;
    hr = oaaNumOfCoinDoors( hDevice, &nNumCoinDoors ) ;
    ... error handling here
    int nCoinDoorNum = 0 ; /* First coin door */
    int nCount = 0 ;
    hr = oaaGetCoinCount( hDevice, nCoinDoorNum, &nCount ) ;
    ... error handling here
    /* use the coin count (modulo-255) here as desired */
    ....

    /* When finished with the device close it */
    hr = oaaCloseDevice( hDevice ) ;
    ... error handling here

    /* Must de-init the library here if we are not going to use it any more */
    hr = oaaDeInit() ;
    ... Any final error handling here

    /* Exit the application*/
}

```

## 4 Building an Application

To build your application using the OAArcade Library you need to include the oaad.h and OAArcadeIF.h header files. Link with the appropriate OAArcade.lib import library file provided.

Install the appropriate OAArcade.dll file in a location that is on your path, or in the same directory that your executable will run from.

Before your application can execute you need to be certain that the OAAD Library has been installed on the target system, as well as any OAAD Device Objects for the devices that you will be interfacing with.

## 5 Limitations

This release of the OAArcade Library implements the event driven mode of operation internally, and expects that your application will poll for data often enough to ensure that the internal modulo-255 counters will not overflow. Refer to the OAArcadeExamp sample application for one way to do this.

Only Coin related inputs, Digital Inputs and Digital Outputs are supported in this release.

Only a single device may be open at one time.

Future releases of this library may add more functionality. If your application requires more functionality than the OAArcade Library provides you will need to use the full OAAO Library for your application. Refer to the documentation and GUIExample sample for more details on how to use the OAAO Library.